NetFRAME

Raptor System

A Bird's Eye View

Karl Johnson (KJ)

Revision 0.99

November 2, 1995

EXHIBIT A

Control Diagnostic and Monitor Subsystem

The control of Raptor is completly "Fly By Wire" - i.e. no physical switch directly controls any function and no indicator is directly controlled by system hardware. All such functions, refered to as Out Of Band functions[1], are controlled through a Control Diagnostic and Monitor (CDM) subsystem implemented by small distributed CDM processors connected on a 400 kbs I^2C serial bus (the CDM Bus). Critical CDM components are powered by the bias power from any of the power supplies, which means that CDM basic CDM functions are avaiable so long as A/C power is available at the input to any of the power supplies. The CDM subsystem supervises or monitors the following system features:

[1]See Glossary

- Power supplies - Presence, status, A/C good, Power On/Off and Output voltages.
- Environment - Ambient and exhaust temperatures, Fan speed, speed control, Fan fault and Overtemp indicators.
- Processor - CPU Presence, Power OK, Overtemp and Fault, NMI control, System reset, Memory type/location and Bus/Core speed ratio.
- I/O - I/O Canister insertion/removal and status indicator, PCI card presence, PCI card power and Smart I/O processor Out Of Band control.
- Historical - Log of all system events, Character mode screen image, and Serial Numbers.

Control of CDM functions is either intrinsic (i.e. CDM components act automatically to perform the function, such as when a fan fails, the remaining fan has its speed increased and the fan failure indicator is lit) or external (i.e. the CDM subsystem gets external input requesting the function). External functions can be initiated from either the system interface port by one of the P6 processors or through the RS232 serial CDM interface connected to the CDM External port.

The CDM subsystem remote access feature provides for remote management of the system when the Operating System is not available[2].

[2]See section on Remote Management.

Raptor Wire Service Architecture

Version 1.0

1/23/96

Prepared for
Raptor Implementation Group

by
Karl Johnson (KJ)

EXHIBIT B

Write Byte Array Message
Request

Slave Address	Type/RW	Start Addr MSB	Start Addr LSB	Length N	Array Data 1	. . .	Array Data N
Check Byte							

Response

Slave Address	Length 0	Status 0/Success	Check Byte

Log Type

The Log data type is to be used for logging byte strings in circular log buffer. It is used to record system events in the NVRAM system log.

Read Log Message
Request

Slave Address	Type/RW	Log Addr MSB	Log Addr LSB	Request 255	Check Byte

Response

Slave Address	Length N+7	Log Time MSB	Log Time	Log Time	Log Time LSB	Log Addr MSB	Log Addr LSB
Log Data Byte 0	. . .	Log Data Byte N	Status 0/Success	Check Byte			

Write Log Message
Request

Slave Address	Type/RW	N/A	N/A	Length N	Log Data Byte 0	. . .	Log Data Byte N
Check Byte							

Response

Slave Address	Length 0	Status 0/Success	Check Byte

The addressing of log entries has some special characteristics.
1) Reading address 65565 (0xffff) is special - It represents the address of the latest entry in the log.
2) Reading address 65564 (0xfffe) is also special - It represents the address of the earliest available entry.

3) The address of real log entries wraps at 65519 (0xffef). The next sequential entry after 65519 is 0.
4) The address of is ignored on write and the next available entry is written.
5) To read the entire log in forward time order, read entry at address 65564. This returns the first log entry along with its actual log address. Increment that address by one and read that entry. Repeat the last step until status indicates failure.
6) To read the entire log in reverse time order, read entry at address 65565. This returns the last log entry along with its actual log address. Decrement that address by one and read that entry. Repeat the last step until status indicates failure.
7)To keep a complete external copy of the log, first read the entire log in forward time order and remember the last valid entry. Then periodically read forward from the remembered last valid entry to the end and add that to the external copy.

Event Type

The event data type is to be used for alerting external interfaces of events in the Wire Service network. Event memory is organized as a queue. The queue will probably be quite small (< 20 Events). Writing an event places the event ID at the next available entry, unless the last queue entry would be written by this event. In that case, the last queue entry is a Queue Overflow Event and the write fails. This allows the external interface to realize that events were lost and it should scan for any changes in data.
Reading the event type returns requested number of events in the queue or the entire queue which ever is less and removes them from the queue.

Read Event Message
Request

Slave Address	Type/RW	N/A	N/A	Request 1-255	Check Byte

Response

Slave Address	Length N	Event ID 1	...	Event ID N	Status 0/Success	Check Byte

Write Event Message
Request

Slave Address	Type/RW	N/A	N/A	Length 1	Event ID	Check Byte

Response

Slave Address	Length 0	Status 0/Success	Check Byte

Possible Event Types:
CPU Status Change
Power Status Change
Canister Status Change
Fan Status Change

Screen Type

The screen data type is to be used for communication of character mode screen information from the system BIOS to remote management interface.

Read Screen Message
Request

Slave Address	Type/RW	S Addr MSB	S Addr LSB	Request 1-255	Check Byte

Response

Slave Address	Length N	Screen Data 1	. . .	Screen Data N	Status 0/Success	Check Byte

Write Screen Message
Request

Slave Address	Type/RW	S Addr MSB	S Addr LSB	Length N	Screen Data 1	. . .	Screen Data n
Check Byte							

Response

Slave Address	Length 0	Status 0/Success	Check Byte

Wire Service Network Physical Connections

The following table describe all of the physical signal connections to all of the Wire Service processors. The names for the connections will be related to network accessible memory data in the section which follows called "Wire Service Network Memory Map".

Note: All signal types and definitions are from the viewpoint of the individual Wire Service PIC processor (e.g. Input means input to PIC processor)

Wire Service System Bus Interface (System Type ID: S0) Processor ID 10

Pin	Type	Name	Function	Notes
RA0	I	S0_FIFO_IEFZ	In FIFO (ISA Writes) Empty Flag (Active Low)	Status Flags for both ISA bus FIFO's. Need to monitor for incoming message overruns. Also must assure that output FIFO is empty before loading output message.
RA1	I	S0_FIFO_IHFZ	In FIFO (ISA Writes) Half-full Flag (Active Low)	
RA2	I	S0_FIFO_IFFZ	In FIFO (ISA Writes) Full Flag (Active Low)	
RA3	I	S0_FIFO_OEFZ	Out FIFO (ISA Reads) Empty Flag (Active Low)	
RA4	I	S0_FIFO_OHFZ	Out FIFO (ISA Reads) Half-full Flag (Active Low)	
RA5	I	S0_FIFO_OFFZ	Out FIFO (ISA Reads) Full Flag (Active Low)	
RB0	I/O	S0_FIFO_D0	ISA FIFOs Data bus Bit 0	This is an 8 bit bi-directional port for the ISA FIFO data bus. These bits should drive the bus before asserting FIFO Write and can be read after tri-stating and asserting FIFO Read
RB1	I/O	S0_FIFO_D1	ISA FIFOs Data bus Bit 1	
RB2	I/O	S0_FIFO_D2	ISA FIFOs Data bus Bit 2	
RB3	I/O	S0_FIFO_D3	ISA FIFOs Data bus Bit 3	
RB4	I/O	S0_FIFO_D4	ISA FIFOs Data bus Bit 4	
RB5	I/O	S0_FIFO_D5	ISA FIFOs Data bus Bit 5	
RB6	I/O	S0_FIFO_D6	ISA FIFOs Data bus Bit 6	
RB7	I/O	S0_FIFO_D7	ISA FIFOs Data bus Bit 7	
RC0	O	S0_FIFO_RZ	FIFO (ISA Writes) Read (Assert Low)	Assert to read the In FIFO
RC1	O	S0_FIFO_WZ	PIC to ISA FIFO (ISA Reads) Write (Assert Low)	Assert to write the Out FIFO
RC2	O	S0_ISA_INT	PIC to ISA Interrupt Request (Assert ?)	Level Interrupt to ISA bus
RC3	I/O	S0_I2C_DATA	Wire Service Bus Clock (I2C)	Only used for I2C
RC4	I/O	S0_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S0_FIFO_RSTZ	In and Out FIFOs Reset (Assert Low)	Resets both FIFOs
RC6	O	S0_FIFO_IRTZ	In FIFO (ISA Writes) Retransmit (Assert Low)	Resets the Read pointer to 0
RC7	O	S0_FIFO_ORTZ	Out FIFO (ISA Reads) Retransmit (Assert Low)	Resets the Read pointer to 0
RD0	I/O	S0_CSR_D0	ISA External Data bus Bit 0 (Slave parallel port)	The slave parallel port is used as a bidirectional control and status register on the ISA bus.
RD1	I/O	S0_CSR_D1	ISA External Data bus Bit 1 (Slave parallel port)	
RD2	I/O	S0_CSR_D2	ISA External Data bus Bit 2 (Slave parallel port)	
RD3	I/O	S0_CSR_D3	ISA External Data bus Bit 3 (Slave parallel port)	
RD4	I/O	S0_CSR_D4	ISA External Data bus Bit 4 (Slave parallel port)	
RD5	I/O	S0_CSR_D5	ISA External Data bus Bit 5 (Slave parallel port)	
RD6	I/O	S0_CSR_D6	ISA External Data bus Bit 6 (Slave parallel port)	
RD7	I/O	S0_CSR_D7	ISA External Data bus Bit 7 (Slave parallel port)	
RE0	I	S0_CSR_RZ	ISA Read Slave Parallel Port (Assert Low)	Not directly manipulated after setting port D/E to act as slave parallel port
RE1	I	S0_CSR_WZ	ISA Write Slave Parallel Port (Assert Low)	
RE2	I	S0_CSR_SZ	ISA Slave Parallel Port Select (Assert Low)	

Wire Service System Monitor A (System Type ID S1) Processor ID 3

Pin	Type	Name	Function	Notes
RA0	O	S1_FAN_HI	System Board fan speed to high (Assert Hi)	Assert on any SB fan failure
RA1	O	S1_SBFAN_LED	System Board fan fault LED	Assert on any SB fan failure
RA2	O	S1_BC_DS0	Bus/Core Speed Ratio and DIMM Select Mux Bit 0	During system reset these bits select bus/core speed ratio for all processors. Otherwise they select which DIMM presents its type on DIMM type port.
RA3	O	S1_BC_DS1	Bus/Core Speed Ratio and DIMM Select Mux Bit 1	
RA4	O	S1_BC_DS2	Bus/Core Speed Ratio and DIMM Select Mux Bit 2	
RA5	O	S1_BC_DS3	Bus/Core Speed Ratio and DIMM Select Mux Bit 3	
RB0	I/O	S1_LCD_D0	LCD Controller Data Bus bit 0	These lines make up the 8 bit data bus to the LCD display
RB1	I/O	S1_LCD_D1	LCD Controller Data Bus bit 1	
RB2	I/O	S1_LCD_D2	LCD Controller Data Bus bit 2	
RB3	I/O	S1_LCD_D3	LCD Controller Data Bus bit 3	
RB4	I/O	S1_LCD_D4	LCD Controller Data Bus bit 4	
RB5	I/O	S1_LCD_D5	LCD Controller Data Bus bit 5	
RB6	I/O	S1_LCD_D6	LCD Controller Data Bus bit 6	
RB7	I/O	S1_LCD_D7	LCD Controller Data Bus bit 7	
RC0	I	S1_FAN_TP	Tachometer pulse input from selected fan	Generally routed to counter
RC1	O?	S1_OK_TO_RUN	Drives SYS_PWRGOOD signal	System starts on 0->1 transition
RC2	I	S1_RESET_SW	Undebounced input from System Reset switch	
RC3	I/O	S1_I2C_DATA	Wire Service Bus Clock (I2C)	Only used for I2C
RC4	I/O	S1_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S1_FAN_SEL0	Fan Tachometer Multiplexer Select Bit 0	Used to select which fan tachometer pulse output is gated to S1_FAN_TP
RC6	O	S1_FAN_SEL1	Fan Tachometer Multiplexer Select Bit 1	
RC7	O	S1_FAN_SEL2	Fan Tachometer Multiplexer Select Bit 2	
RD0	I	S1_DIMM_D0	DIMM Type port bit 0	These lines make up an 8 bit port which on which the DIMM module in the slot selected by S1_BC_DS0..3 presents its type data if any. If no DIMM is present in the slot selected the DIMM type bits are all 1's.
RD1	I	S1_DIMM_D1	DIMM Type port bit 1	
RD2	I	S1_DIMM_D2	DIMM Type port bit 2	
RD3	I	S1_DIMM_D3	DIMM Type port bit 3	
RD4	I	S1_DIMM_D4	DIMM Type port bit 4	
RD5	I	S1_DIMM_D5	DIMM Type port bit 5	
RD6	I	S1_DIMM_D6	DIMM Type port bit 6	
RD7	I	S1_DIMM_D7	DIMM Type port bit 7	
RE0	O	S1_LCD_RS	LCD Controller Register Select	See LCD Controller data sheet for details of operation of these signals
RE1	O	S1_LCD_ENA	LCD Controller Register Enable	
RE2	O	S1_LCD_RW	LCD Controller Register Read/Write	

Wire Service System Monitor B (System Type ID S2) Processor ID 4

Pin	Type	Name	Function	Notes
RA0	O	S2_FLASH_LED	CPU Display the Enable/Disable state of the BIOS Flash ROM	Should track S2_FLASH_ENA
RA1	O	S2_SBFLT_LED0	System Board FRU LED Pin 0 (bicolor LED)	Drive in different combinations for OFF, AMBER, GREEN
RA2	O	S2_SBFLT_LED1	System Board FRU LED Pin 1 (bicolor LED)	
RA3	O	S2_OVRTMP_LED	Over Temperature LED	
RA4	I	S2_TEMP_CPU4	Thermal Fault - CPU 4	Indicator that CPU has exceeded temperature limit and faulted
RA5	I	S2_TEMP_CPU3	Thermal Fault - CPU 3	
RB0	O	S2_SB_JTAG	Enable System Board JTAG Chain TMS	
RB1	O	S2_FLASH_WE	System BIOS FLASH Write Enable	
RB2	I	S2_FLASH_SW	System BIOS FLASH Write Enable Switch (undebounced)	
RB3	I	S2_NMI_SW	System Non-Maskable Interrupt (NMI) Switch (undebounced)	
RB4	I	S2_POK_CPU1	Power Good signal from CPU 1	Indicator that power regulator for CPU is operating correctly. Only valid if corresponding CPU is present (S2_PRES_CPUx)
RB5	I	S2_POK_CPU2	Power Good signal from CPU 2	
RB6	I	S2_POK_CPU3	Power Good signal from CPU 3	
RB7	I	S2_POK_CPU4	Power Good signal from CPU 4	
RC0	x		Unused	
RC1	x		Unused	
RC2	O	S2_NMI_CPU4	NMI Request for CPU 4	Toggle to cause NMI to CPU
RC3	I/O	S2_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S2_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S2_NMI_CPU3	NMI Request for CPU 3	See S2_NMI_CPU4 above
RC6	O	S2_NMI_CPU2	NMI Request for CPU 2	
RC7	O	S2_NMI_CPU1	NMI Request for CPU 1	
RD0	I	S2_PRES_CPU1	Presence detection bit - CPU 1	Asserted when a processor inserted in the system board
RD1	I	S2_PRES_CPU2	Presence detection bit - CPU 2	
RD2	I	S2_PRES_CPU3	Presence detection bit - CPU 3	
RD3	I	S2_PRES_CPU4	Presence detection bit - CPU 4	
RD4	I	S2_ERROR_CPU1	Processor Fault bit - CPU 1	Processor either failed BIST on startup or later other fault. Only valid if corresponding CPU is present (S2_PRES_CPUx)
RD5	I	S2_ERROR_CPU2	Processor Fault bit - CPU 2	
RD6	I	S2_ERROR_CPU3	Processor Fault bit - CPU 3	
RD7	I	S2_ERROR_CPU4	Processor Fault bit - CPU 4	
RE0	O	S2_SYSFLT_LED	System Fault summary LED	
RE1	I	S2_TEMP_CPU2	Thermal Fault - CPU 2	See S2_TEMP_CPU4 above
RE2	I	S2_TEMP_CPU1	Thermal Fault - CPU 1	

Wire Service System Recorder (System Type ID S3) Processor ID 1

Pin	Type	Name	Function	Notes
RA0	O	S3_NVRAM_A8	NVRAM Address Bit 8	NVRAM Address Bus Bits 8-13
RA1	O	S3_NVRAM_A9	NVRAM Address Bit 9	
RA2	O	S3_NVRAM_A10	NVRAM Address Bit 10	
RA3	O	S3_NVRAM_A11	NVRAM Address Bit 11	
RA4	O	S3_NVRAM_A12	NVRAM Address Bit 12	
RA5	O	S3_NVRAM_A13	NVRAM Address Bit 13	
RB0	I/O	S3_NVRAM_D0	NVRAM Data Bit 0	NVRAM 8 Bit Data Bus
RB1	I/O	S3_NVRAM_D1	NVRAM Data Bit 1	
RB2	I/O	S3_NVRAM_D2	NVRAM Data Bit 2	
RB3	I/O	S3_NVRAM_D3	NVRAM Data Bit 3	
RB4	I/O	S3_NVRAM_D4	NVRAM Data Bit 4	
RB5	I/O	S3_NVRAM_D5	NVRAM Data Bit 5	
RB6	I/O	S3_NVRAM_D6	NVRAM Data Bit 6	
RB7	I/O	S3_NVRAM_D7	NVRAM Data Bit 7	
RC0	O	S3_NVRAM_CSZ	NVRAM Chip Select (Negative Logic)	Control signals for NVRAM - See Dallas DS1245 data sheet
RC1	O	S3_NVRAM_OEZ	NVRAM Output Enable (Negative Logic)	
RC2	O	S3_NVRAM_WEZ	NVRAM Write Enable (Negative Logic)	
RC3	I/O	S3_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S3_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S3_NVRAM_A14	NVRAM Address Bit 14	NVRAM Address Bus Bits 14-16
RC6	O	S3_NVRAM_A15	NVRAM Address Bit 15	
RC7	O	S3_NVRAM_A16	NVRAM Address Bit 16	
RD0	O	S3_NVRAM_A0	NVRAM Address Bit 0	NVRAM Address Bus Bits 0-7
RD1	O	S3_NVRAM_A1	NVRAM Address Bit 1	
RD2	O	S3_NVRAM_A2	NVRAM Address Bit 2	
RD3	O	S3_NVRAM_A3	NVRAM Address Bit 3	
RD4	O	S3_NVRAM_A4	NVRAM Address Bit 4	
RD5	O	S3_NVRAM_A5	NVRAM Address Bit 5	
RD6	O	S3_NVRAM_A6	NVRAM Address Bit 6	
RD7	O	S3_NVRAM_A7	NVRAM Address Bit 7	
RE0	O	S3_RTC_CLK	Real Time Clock - Data Clock	See Dallas DS1603 data sheet
RE1	I	S3_RTC_DATA	Real Time Clock - Serial Data	
RE2	O	S3_RTC_RSTZ	Real Time Clock - Protocol Reset (Negative Logic)	

Wire Service Backplane (System Type ID S4) Processor ID 2

Pin	Type	Name	Function	Notes
RA0	A	S4_VOLTS_P5V	Analog measure of system +5 volt main supply	Use D/A converter to read voltages as 0-255. Calibration constants are determined externally
RA1	A	S4_VOLTS_P3V	Analog measure of system +3.3 volt main supply	
RA2	A	S4_VOLTS_P12V	Analog measure of system +12 volt main supply	
RA3	A	S4_VREF	Voltage Reference for A/D converter	Unused
RA4	x			
RA5	A	S4_VOLTS_N12V	Analog measure of system -12 volt main supply	See S4_VOLTS_P5V
RB0	I/O	S4_PSN_CAN1	Presence and Serial Number I/O for Canister 1	These are all lines to one wire serial data EPROMS. See Dallas DS250x data sheet for programming information
RB1	I/O	S4_PSN_CAN2	Presence and Serial Number I/O for Canister 2	
RB2	I/O	S4_PSN_CAN3	Presence and Serial Number I/O for Canister 3	
RB3	I/O	S4_PSN_CAN4	Presence and Serial Number I/O for Canister 4	
RB4	I/O	S4_PSN_CAN5	Presence and Serial Number I/O for Canister 5	
RB5	I/O	S4_PSN_CAN6	Presence and Serial Number I/O for Canister 6	
RB6	I/O	S4_PSN_CAN7	Presence and Serial Number I/O for Canister 7	
RB7	I/O	S4_PSN_CAN8	Presence and Serial Number I/O for Canister 8	
RC0	x			
RC1	I	S4_ACOK_PS3	A/C Input OK to Power Supply 3	Should check only if PSN for power supply indicates presence.
RC2	I	S4_ACOK_PS2	A/C Input OK to Power Supply 2	
RC3	I/O	S4_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S4_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	I	S4_ACOK_PS1	A/C Input OK to Power Supply 1	See S4_ACOK_PS3
RC6	O	S4_POWER_ON	Enable main output from power supplies	
RC7	I	S4_POWER_SW	Power On/Off switch (undebounced)	
RD0	I/O	S4_PSN_PS1	Presence and Serial Number for Power Supply 1	These are all lines to one wire serial data EPROMS. See Dallas DS250x data sheet
RD1	I/O	S4_PSN_PS2	Presence and Serial Number for Power Supply 2	
RD2	I/O	S4_PSN_PS3	Presence and Serial Number for Power Supply 3	
RD3	I/O	S4_PSN_BP	Presence and Serial Number for Backplane	Dallas DS250x also
RD4	I/O	S4_PSN_SB	Presence and Serial Number for System Board	Dallas DS250x also
RD5	I	S4_BP_TYPE	Backplane Type (0: Small 1: Large)	
RD6	I/O	S4_TEMP_SCL	Temperature Bus Clock	I2C local bus for temperature probes at different system points
RD7	I/O	S4_TEMP_SDA	Temperature Bus Serial Data	
RE0	I	S4_DCOK_PS3	D/C Output OK from Power Supply 3	Should check only if PSN for power supply indicates presence.
RE1	I	S4_DCOK_PS2	D/C Output OK from Power Supply 2	
RE2	I	S4_DCOK_PS1	D/C Output OK from Power Supply 1	

Wire Service Canister (System Type ID S5) Processor ID 2x where x is the slot ID

Pin	Type	Name	Function	Notes
RA0	O	S5_P12V_ENA	Turns on +/- 12 volt to all PCI slots	Used to sequence power to PCI cards.
RA1	O	S5_P5V_ENA4	Turns on +5 volts to PCI slot 4	
RA2	O	S5_P5V_ENA3	Turns on +5 volts to PCI slot 3	
RA3	O	S5_P5V_ENA2	Turns on +5 volts to PCI slot 2	
RA4	O	S5_P5V_ENA1	Turns on +5 volts to PCI slot 1	
RA5	?			
RB0	I	S5_CAN_A0	Canister Address bit 0	Determine the Wire Service bus address of this canister
RB1	I	S5_CAN_A1	Canister Address bit 1	
RB2	I	S5_CAN_A2	Canister Address bit 2	
RB3	I	S5_PRSNT_S5	Special Slot 5 (IOP/PCI jumper) present	Indicates something is in slot 5
RB4	I/O	S5_PSN_S5	Present Serial Number for special slot 5	From DS 250x in slot 5 card (if IOP)
RB5	x			
RB6	x			
RB7	x			
RC0	I	S5_FAN_TP	Tachometer pulse input from selected fan	Generally routed to counter
RC1	O	S5_FAN_SEL0	Fan Tachometer Multiplexer Select Bit 0	Select which fan to monitor tach.
RC2	x			
RC3	I/O	S5_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S5_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S5_CANFAN_LED	Canister fan fault LED	Assert on any Canister fan failure
RC6	O	S5_CANFLT_LED0	Canister FRU LED Pin 0 (bicolor LED)	Drive in different combinations for OFF, AMBER, GREEN
RC7	O	S5_CANFLT_LED1	Canister FRU LED Pin 1 (bicolor LED)	
RD0	I	S5_PRSNT_S1A	PCI card present in Slot 1 (A pin)	PCI slots have 2 presence pins - see PCI spec for usage and meaning.
RD1	I	S5_PRSNT_S1B	PCI card present in Slot 1 (B pin)	
RD2	I	S5_PRSNT_S2A	PCI card present in Slot 2 (A pin)	
RD3	I	S5_PRSNT_S2B	PCI card present in Slot 2 (B pin)	
RD4	I	S5_PRSNT_S3A	PCI card present in Slot 3 (A pin)	
RD5	I	S5_PRSNT_S3B	PCI card present in Slot 3 (B pin)	
RD6	I	S5_PRSNT_S4A	PCI card present in Slot 4 (A pin)	
RD7	I	S5_PRSNT_S4B	PCI card present in Slot 4 (B pin)	
RE0	O	S5_CAN_JTAG	Enable Canister Board JTAG Chain TMS	Required to select the JTAG chain
RE1	O	S5_NMI_S5	NMI card is special slot 5 (IOP)	Toggle to NMI IOP in slot 5
RE2	O	S2_FAN_HI	Canister fan speed to high (Assert Hi)	Assert on any Canister fan failure

Wire Service Remote Interface (System Type ID S6) Processor ID 11

Pin	Type	Name	Function	Notes
RA0	I/O	S6_PSN_RI	Serial Number information for Remote Interface	
RA1	x			
RA2	x			
RA3	x			
RA4	x			
RA5	x			
RB0	O	S6_MODEM_DTR	Modem Signal (Data Terminal Ready)	
RB1	I	S6_MODEM_DSR	Modem Signal (Data Set Ready)	
RB2	I	S6_MODEM_CD	Modem Signal (Carrier Detect)	
RB3	I	S6_MODEM_RI	Modem Signal (Ring Indicate)	
RB4	O	S6_MODEM_RTS	Modem Signal (Request To Send)	
RB5	I	S6_MODEM_CTS	Modem Signal (Clear To Send)	
RB6	x			
RB7	x			
RC0	x			
RC1	x			
RC2	x			
RC3	I/O	S5_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S5_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	x			
RC6	O	S6_MODEM_TXD	Modem Signal (Transmit Data)	Controlled by chip serial interface
RC7	I	S6_MODEM_RXD	Modem Signal (Receive Data)	
RD0	x			
RD1	x			
RD2	x			
RD3	x			
RD4	x			
RD5	x			
RD6	x			
RD7	x			
RE0	x			
RE1	x			
RE2	x			

Wire Service Network Memory Map

This section defines the Wire Service Network Memory Map for the first Raptor system. Its purpose is to identify all Wire Service addressable entities and describe their function and any special information about them.

This section is incomplete yet and only a small incomplete sample is supplied. (Although some of the more complicated ones are described.)

The address format is "pp:aaaa", where "p" is the processor ID (hexadecimal) of the Wire Service Processor where the data resides and "aaaa" is the hexadecimal address or address range for the data.

Name	Type	Address	Description	Notes
WS_DESC_Pn	STRING	0n:0000	Wire Service Processor Type/Description	
WS_REV_Pn	STRING	0n0001	Wire Service Software Revision/Date Info	
WS_SB_FAN_HI	BIT	03	System Board Fans HI	Controls S1_FAN_HI. Set on 0->1 transition of WS_SB_FAN_LED. Cleared by other software
WS_SB_FAN_LED	BIT	03	System Board Fan Fault LED	Controls S1_SBFAN_LED. It is set whenever any WS_SB_FANFAULTn is set. Log 0->1 transition
WS_SB_BUSCORE	BYTE	03	System Board BUS/CORE speed ratio to use on reset	Value is asserted on S1_BC_DS[0-3] unless reading DIMM types. Set to 0 on power on.
WS_SYS_LCD	STRING	03	Value to display on LCD	For a Nx2 display the first N bytes display on top line and the second N bytes display on the bottom line. Manipulates S1_LCD_D[0-7], S1_LCD_RS, S1_LCD_ENA, S1_LCD_RW
WS_SB_FAN1	BYTE	03	System Board Fan 1 speed	Approximately every second a fan is selected by S1_FAN_SEL[0-2] and monitored via S1_FAN_TP driving a counter for a known period of time. The counter is then loaded into the appropriate fan speed. If WS_SB_FAN_HI is not set then the speed is compared against WS_SB_FAN_LOLIM. If fan is slow set appropriate WS_SB_FANFAULTn otherwise clear it
WS_SB_FAN2	BYTE	03	System Board Fan 2 speed	
WS_SB_FAN3	BYTE	03	System Board Fan 3 speed	
WS_SB_FAN4	BYTE	03	System Board Fan 4 speed	
WS_SB_FANFAULT1	BIT	03	System Board Fan 1 Faulted	
WS_SB_FANFAULT2	BIT	03	System Board Fan 2 Faulted	

WS_SB_FANFAULT3	BIT	03	System Board Fan 3 Faulted	
WS_SB_FANFAULT4	BIT	03	System Board Fan 4 Faulted	
WS_SB_FAN_LOLIM	BYTE	03	Fan speed low speed fault limit	Set to ??? on power on
WS_SB_DIMM_SEL	BYTE	03	The DIMM select bits to use when reading DIMM_TYPE	The low order 4 bits are the select bits to use when WS_SB_DIMM_TYPE is read.
WS_SB_DIMM_TYPE	BYTE	03	The type of DIMM in the DIMM_SEL position	When read asserts value of WS_SB_DIMM_SEL on S1_BC_DS[0-3] and then returns value of S1_DIMM_D[0-7]
WS_SB_FLASH_ENA	BIT	04	Indicates FLASH ROW write enabled	Set/Cleared by debounced 0->1 transition of S2FLASH_SW. Controls state of S2_FLASH_WE and S2_FLASH_LED.
WS_SB_FRU_FAULT	BIT	04	Indicates the FRU status	At power on starts at 1. Controls S2_SBFLT_LED[0-1] for bicolor LED colors 0=Green 1=Amber. Cleared by other software
WS_SYS_OVERTEMP	BIT	04	Indicates Overtemp fault	At power on is set. Controls S2_OVRTMP_LED. Controlled by wire service backplane processor.
WS_SB_JTAG	BIT	04	Enables JTAG chain on system board	Clear at power on. Controls S2_SB_JTAG
WS_SB_CPU_PRES	BYTE	04	CPU Presence bits (LSB = CPU1)	Assemble from S2_PRES_CPU[1-4]
WS_SB_CPU_ERR	BYTE	04	CPU Error bits (LSB = CPU1)	Assemble from S2_ERROR_CPU[1-4]
WS_SB_CPU_TEMP	BYTE	04	CPU Thermal fault bits (LSB = CPU1)	Assemble from S2_TEMP_CPU[1-4]
WS_SB_CPU_POK	BYTE	04	CPU Power OK (LSB = CPU1)	Assemble from S2_POK_CPU[1-4]
WS_NMI_MASK	BYTE	04	CPU NMI processor mask (LSB=CPU1)	Defaults to all ones on power up
WS_NMI_REQ	BIT	04	NMI Request bit	When set pulse S2_NMI_CPUn corresponding to each bit set in WS_NMI_MASK. Then clear request bit. Log Action
WS_SYSFAULT	BIT	04	System Fault Summary	This bit is set if any faults detected in the system. Controls S2_SYSFLT_LED Bits scanned WS_SP_CPU_FAULT, WS_SB_FRU_FAULT (other faults?)
WS_SB_CPU_FAULT	BIT	04	CPU Fault Summary	This bit is set if ((WS_SB_CPU_ERR | WS_SB_CPU_TEMP | ~WS_SB_CPU_POK) & ~WS_SB_CPU_PRES) != 0. Log 0->1 transition with CPU bytes.

WS_BP_P5V	BYTE	02	Analog Measure of +5 volt main supply	read from S4_VOLTS_P5v
WS_BP_P3V	BYTE	02	Analog Measure of +3.3 volt main supply	read from S4_VOLTS_P3v
WS_BP_P12V	BYTE	02	Analog Measure of +12 volt main supply	read from S4_VOLTS_P12v
WS_BP_P5V	BYTE	02	Analog Measure of -12 volt main supply	read from S4_VOLTS_N12V
WS_SYS_CAN_PRES	BYTE	02	Presence bits for canisters (LSB=1, MSB=8)	controlled by S4_PSN_CAN[1-8]. A previous value byte needs to be maintained so canister transitions can be recognized. Previous value initialized to zero. Periodic monitor scans for new canisters. When new canister is recognized read full serial data and store in WS_SYS_CAN_SERIALn then log and send event
WS_SYS_PS_PRES	BYTE	02	Presence bits for power supplies (LSB=1, MSB=3)	controlled by S4_PSN_PS[1-3]. A previous value byte needs to be maintained so power supply transitions can be recognized. Previous value initialized to zero. Periodic monitor scans for new power supplies. When new power supply is recognized read full serial data and store in WS_SYS_PS_SERIALn then log and send event
WS_SYS_PS_ACOK	BYTE	02	Power supply ACOK status (LSB=1, MSB=3)	controlled by S4_ACOK_PS[1-3]. A previous value byte needs to be maintained so power supply transitions can be recognized. Previous value initialized to zero. Periodic monitor scans for changes is ACOK and sends events
WS_SYS_PS_DCOK	BYTE	02	Power supply DCOK status (LSB=1, MSB=3)	controlled by S4_DCOK_PS[1-3]. A previous value byte needs to be maintained so power supply transitions can be recognized. Previous value initialized to zero. Periodic monitor scans for changes is ACOK and sends events
WS_SYS_BP_TYPE	BYTE	02	Type of system backplane currently only two types Type 0 = 4 canister (small) and Type 1= 8 canister (large)	controlled by S4_BP_TYPE
WS_SYS_TEMP_SB1	BYTE	02	Temperature of system board position 1	controlled by reading Dallas temperature transducers connected to serial bus on S4_TEMP_SDA and S4_TEMP_SCL
WS_SYS_TEMP_SB2	BYTE	02	Temperature of system board position 2	
WS_SYS_TEMP_BP1	BYTE	02	Temperature of backplane position 1	
WS_SYS_TEMP_BP2	BYTE	02	Temperature of backplane position 2	

WS_SYS_TEMP_WARN	BYTE	02	Warning temperature. Initialized to ???	If any WS_SYS_TEMP_xxn exceeds this value, log, send event, set WS_SYS_OVERTEMP
WS_SYS_TEMP_SHUT	BYTE	02	Shutdown temperature. Initialized to ???	If any WS_SYS_TEMP_xxn exceeds this value, log and clear WS_SYS_POWER
WS_SYS_REQ_POWER	BIT	02	Set to request main power on	
WS_SYS_POWER	BIT	02	Controls system master power S4_POWER_ON	When this bit is set 0->1 set S4_POWER_ON, WS_SYS_RUN = 0, WS_SYS_RSTIMER = 5 and log. When this bit is cleared clear S4_POWER_ON and log.
WS_SYS_RSTIMER	BYTE	02	Used to delay reset/run until power stabilized	Counts down to 0 at 10 counts per second. When 1->0 transition sets WS_SYS_RUN.
WS_SYS_RUN	BIT	02	Controls the system halt/run line S1_OK_TO_RUN.	If this bit is cleared, clear S1_OK_TO_RUN and log. If this bit is set, set S1_OK_TO_RUN and log.
WS_CAN_POWER	BIT	2x	Controls canister PCI slot power	When set then set S5_P5V_ENA[1..4],S5_P12V_ENA in that order with small (about 1 ms) delay between each, then log. When cleared then clear S5_P12V_ENA, S5_P5V_ENA[1..4] then log
WS_CAN_PCI_PRESENT	BYTE	2x	Reflects PCI card slot[1..4] presence indicator pins (MSB to LSB) 4B,4A,3B,3A,2B,2A,1B, 1A	Reflects data from S5_PRSNT_S[1..4][A/B]
WS_CAN_S5_PRESENT	BIT	2x	Indicates the presence of something in slot 5	Reflects S5_PRSNT_S5
WS_CAN_S5_SMART	BIT	2x	Indicates something other than a passive board in slot 5	On power up attempt to read Dallas serial number chip using S5_PSN_S5. If present set this bit and read full serial data and store in WS_SYS_CAN_IOP_SERIALn
WS_CAN_FAN_HI	BIT	2x	Canister Fans HI	Controls S1_FAN_HI. Set on 0->1 transition of WS_SB_FAN_LED. Cleared by other software
WS_CAN_FAN_LED	BIT	2x	Canister Fan Fault LED	Controls S5_CANFAN_LED. It is set whenever any WS_CAN_FANFAULTn is set. Log 0->1 transition
WS_CAN_FANFAULT1	BIT	03	Canister Fan 1 Faulted	
WS_CAN_FANFAULT2	BIT	03	Canister Fan 2 Faulted	

WS_CAN_FAN1	BYTE	2x	Canister Fan 1 speed	Approximately every second a fan is selected by S5_FAN_SEL0 and monitored via S5_FAN_TP driving a counter for a known period of time. The counter is then loaded into the appropriate fan speed. If WS_CAN_FAN_HI is not set then the speed is compared against WS_CAN_FAN_LOLIM. If fan is slow set appropriate WS_CAN_FANFAULTn otherwise clear it
WS_CAN_FAN2	BYTE	2x	Canister Fan 2 speed	
WS_CAN_FAN_LOLIM	BYTE	2x	Fan low speed fault limit	Set to equivalent of xxx RPM on power on
WS_CAN_JTAG_ENA	BIT	2x	Enable JTAG TMS chain for canister	Copy set value to S5_CAN_JTAG
WS_CAN_NMI_S5	BIT	2x	NMI card in slot 5	when set, pulse S2_NMI_S5
WS_RI_CD	BIT	11	Status of Remote Port Modem CD	Follows S6_MODEM_CD
WS_RI_DTR	BIT	11	State of Remote Port Modem DTR	Controls S6_MODEM_DTR
WS_RI_DSR	BIT	11	Status of Remote Port Modem DSR	Follows S6_MODEM_DSR
WS_RI_RTS	BIT	11	Status of Remote Port Modem RTS	Controls S6_MODEM_RTS
WS_RI_CTS	BIT	11	Status of Remote Port Modem CTS	Follows S6_MODEM_CTS
WS_RI_CALLOUT	BYTE	11	Controls Call out Script activation	If written to it initiates Call out sequence programmed in WS_SYS_CALL_SCRIPT passing value as argument to script. Log it (Format of Script Programs TBD)
WS_RI_EVENTS	EVENT	11	Remote Interface Event Queue	See Event Data type description in prior section.
WS_SI_EVENTS	EVENT	10	System Interface Event Queue	See Event Data type description in prior section.
WS_SYS_LOG	LOG	01	System Log	The system log kept in NVRAM (See LOG data type in previous section)
WS_SYS_SCREEN	SCREEN	01	System Screen	A copy of the most recent character mode screen from the system video display (See SCREEN data type in previous section)
WS_SYS_SB_SERIAL	STRING	01	Last known System Board serial data	
WS_SYS_BP_SERIAL	STRING	01	Last known Back Plane serial data	
WS_SYS_RI_SERIAL	STRING	01	Last known Remote Interface serial data	

WS_SYS_CAN_SERIAL[1-8]	STRING	01	Last known Canister [1-8] Serial data	May be zero length if no canister ever seen
WS_SYS_IOP_SERIAL[1-8]	STRING	01	Last known IOP in Canister [1-8] Serial data	May be zero length if no canister ever seen or current canister has no IOP
WS_SI_QUEUE	QUEUE	01	Queue of data going to System Interface	See Queue data type in previous section
WS_RI_QUEUE	QUEUE	01	Queue of data going to Remote Interface	See Queue data type in previous section
WS_SYS_XDATA	BYTE ARRAY	01	Byte Array for storage of arbitrary external data in NVRAM	Wire Service just maintains this data area and is unaware of the meaning of any data stored in it.
WS_SYS_EXT_KB	BYTE	01	Size of the WS_SYS_XDATA in kilobytes	Necessary for memory management of the data area

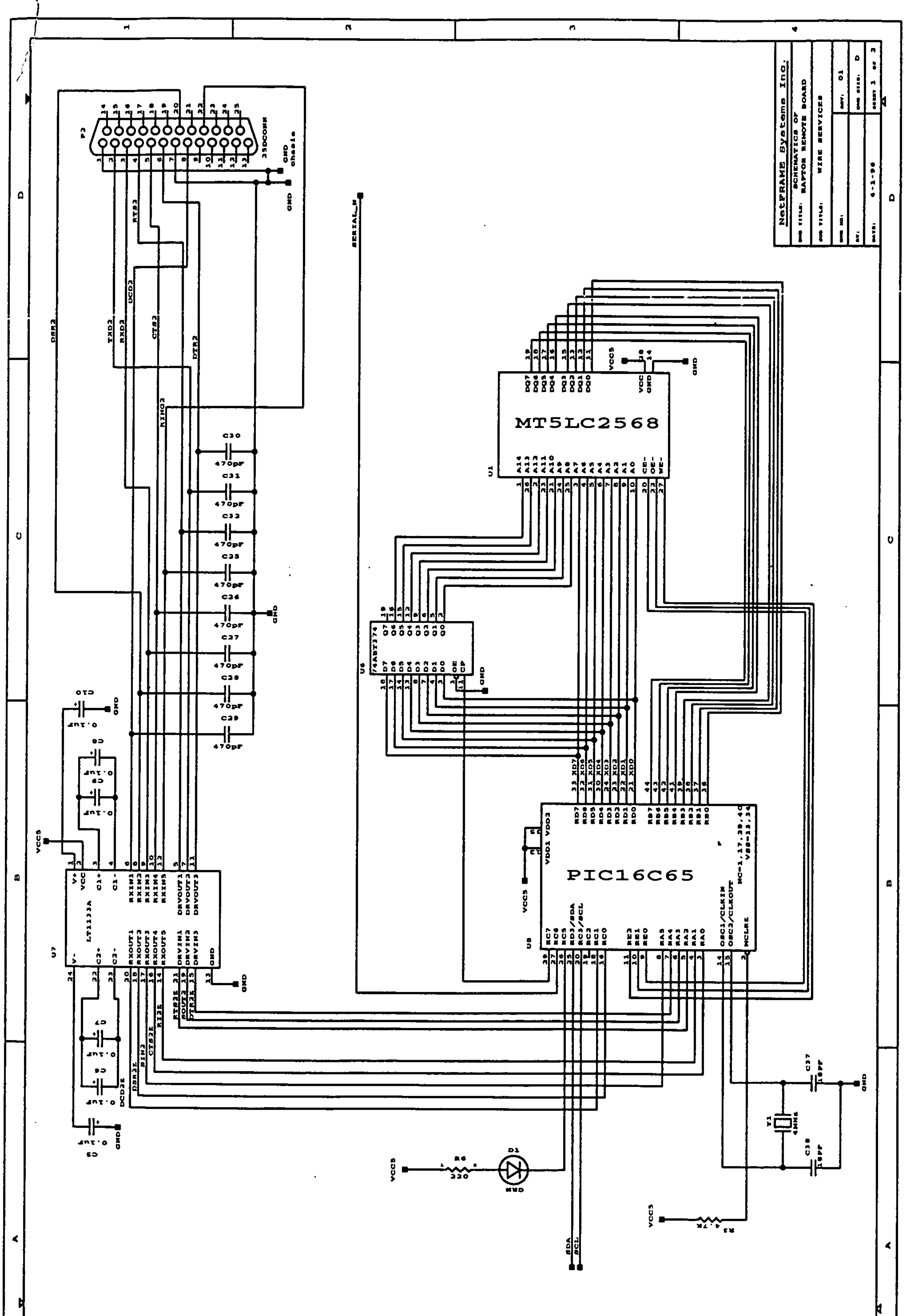
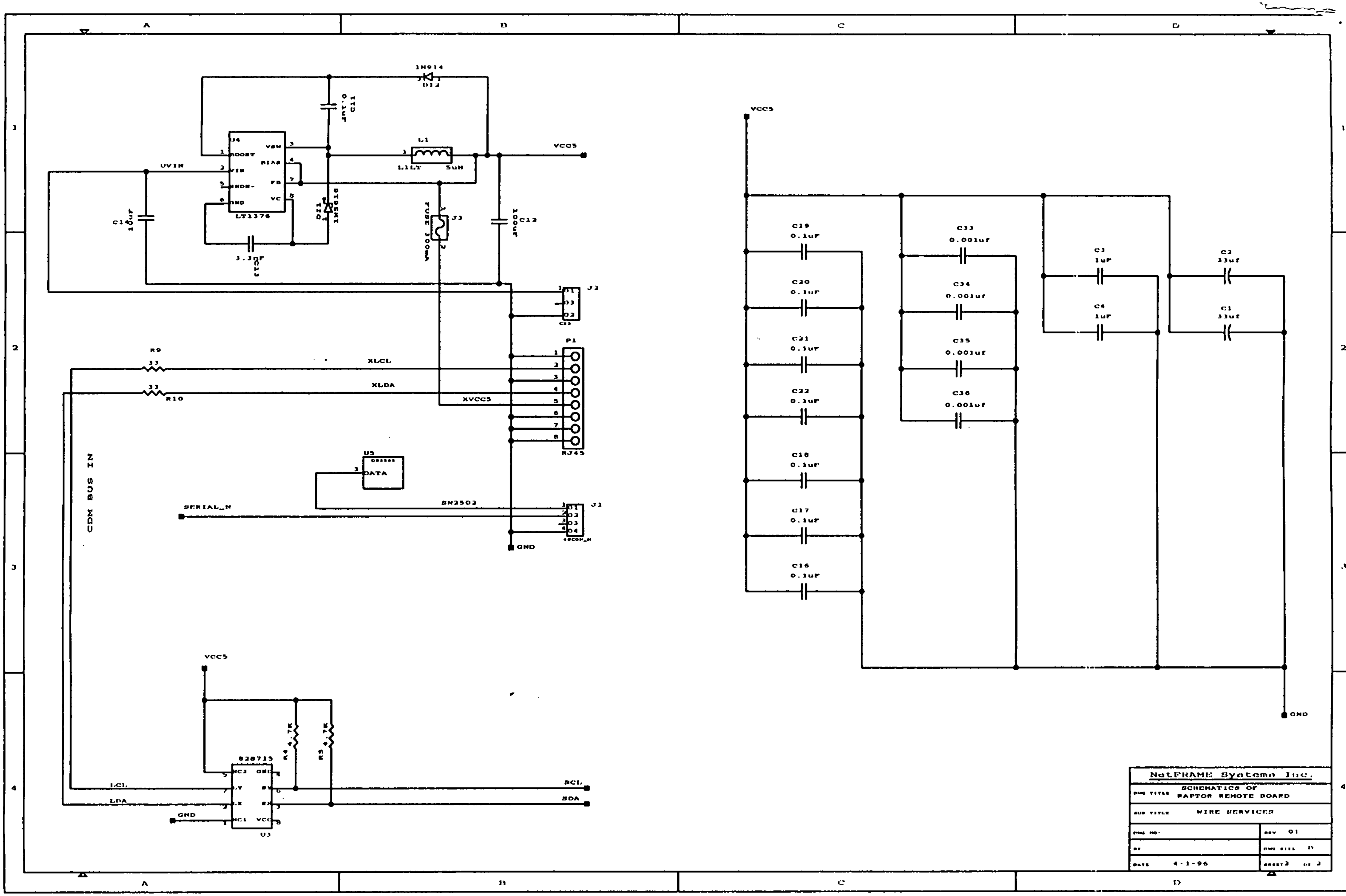

NetFRAME Systems Inc.
SCHEMATICS OF RAPTOR REMOTE BOARD
WIRE SERVICES
4-1-96
MT5LC2568
PIC16C65
LT1133A
74ABT374
25DCONN
SERIAL_N
SDA
SCL
470pF
4MHz
220
4.7K

EXHIBIT C



NetFRAME Systems Inc.
DWG TITLE SCHEMATICS OF RAPTOR REMOTE BOARD
SUB TITLE WIRE SERVICES
DWG NO.
REV 01
BY
DWG SIZE D
DATE 4-1-96
SHEET 2 OF 2
VCC5
GND
U4
LT1376
BOOST
VIN
SHDN-
GND
VSW
BIAS
FB
VC
UVIN
L1
LILT
5uH
1N914
D12
C11
0.1uF
C12
100uF
C13
3.3nF
C14
10uF
D11
1N5818
J3
FUSE 300mA
J2
P1
RJ45
XLCL
XLDA
XVCC5
R9
33
R10
U5
DATA
SERIAL_N
SN2502
J1
CDM BUS IN
U3
828715
LCL
LDA
SCL
SDA
R4
4.7K
R5
4.7K
C19
C20
C21
C22
C18
C17
C16
0.1uF
C33
C34
C35
C36
0.001uF
C3
C4
1uF
C2
C1
33uf

Remote

Interface Board

Specification

Revision 2
13-000072-01
June 21, 1996

Tahir Sheikh
NetFRAME Systems, Inc.

Index

Overview:

This board is an interface between Raptor Wire Services and an external modem. The system status and commands are passed through the RS232 connection at the modem side to the Wire Services bus, the I2C bus, controlled through an on board PIC16C65. The I2C signals are translated by the PIC16C65 into an eight signal RS232 protocol and passed through a voltage level translator LT1133A, with baud capable of reaching the speed of 120k. A 25 pin D-Sub connector resides on the other side of the voltage level translator.

The system status storage is through a 32Kx8 SRAM, with an external lath for latching the higher addressing bits of the data RAM. A signal powered EPROM is used for storing board ID information.

The board is powered through 7.5V and 700mA supply unit, and is an alternative source for the bias powered partition of the Wire Services. The bias powered block includes an NV-RAM and a PIC16C65 which are resident on the Raptor back plane. The power source is regulated through a high frequency switching regulator.

1.0 Features

The designed features are as follows:

1.1 I2C Interface

The two wires interface is brought from the Raptor and passed to the PIC16C65 using an RJ45. A bus extender 82B715 is connected between the external interface to the local I2C bus. Port C bit 3 is the clocking bit, and Port C bit 4 is the data line.

1.2 RS232 Protocol

The communication with the modem is based on the RS232. Microcontroller PIC16C65 is used to generate the receive and the transmit signals, where the signal levels are transposed to the RS232 levels by the LT1133A. The 3 transmit signals, RTS, SOUT and DTR are from Port A bits 2, 3 and 4, where as the 5 receive signals are from two ports, DCD, DSR from Port C 1,0 and SIN, CTS and RI from Port A 5, 0, 1.

The 25 pin RS232 pin connection is used instead a 9 pin connector, since this type of connector is more common than the other. All the extra pins are no connect except the pins 1 and 7, where pin 1 is chassis ground and pin 7 is a signal ground.

The connection through LT1133A can be run up to 120k Baud and is ESD protected to +/- 10kV.

The short voltage at the output can be +/- 30V and is isolated to the forward direction only.

1.3 PIC16C65 and 32Kx8

A 32Kx8 SRAM is available for storage and transfer between the internal Wire Services and the external remote interface. Port D is the address port, while an external 74ABT374 is for expanding the address range to 15 bits. Port B is the data bus for the bi-directional data interconnect. Port E is for the SRAM enable, output tristate and the write control signals.

The PIC16C65 is designed for a frequency of 12MHz. An LED is also connected to the Port C bit 5.

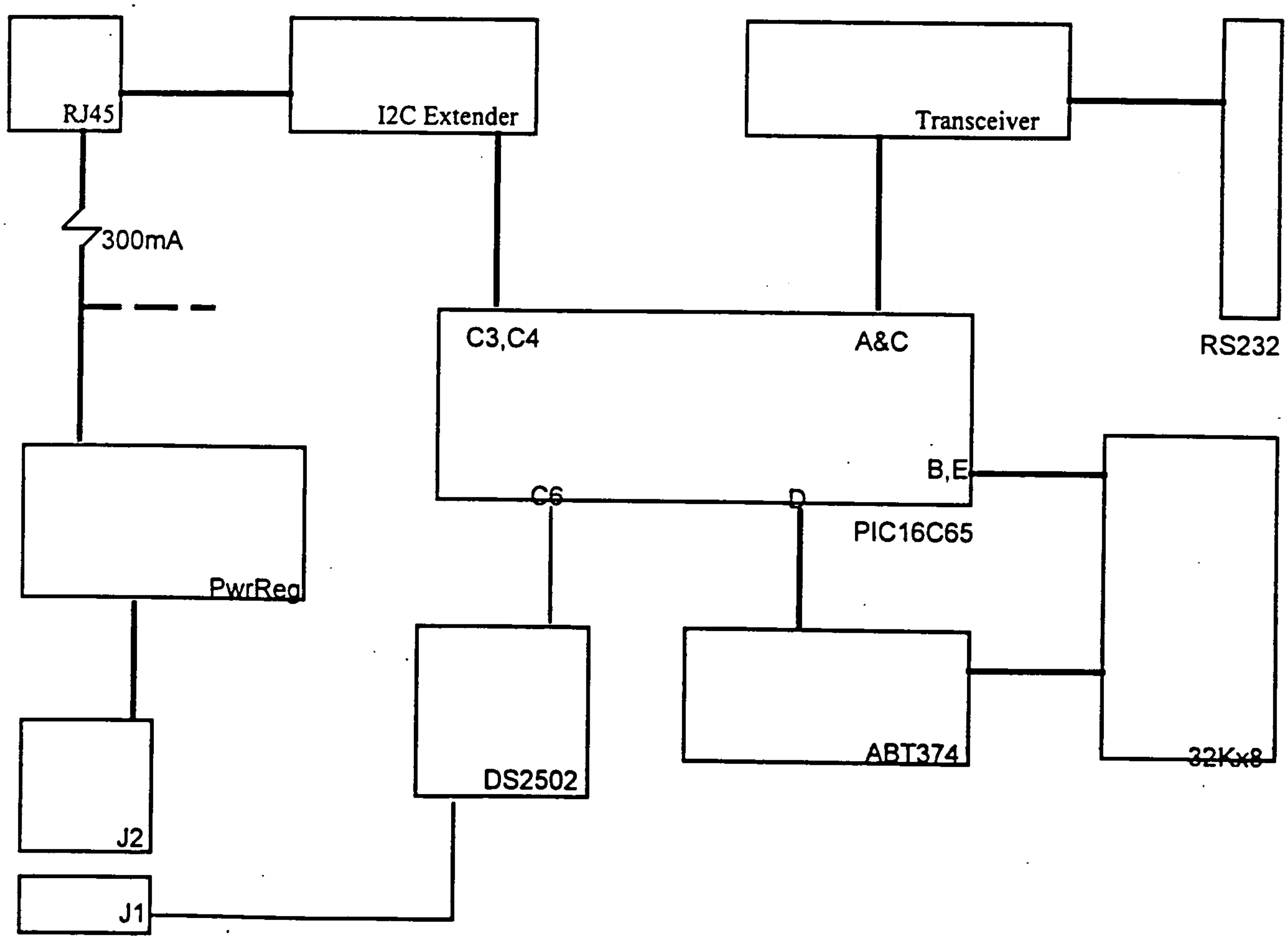


Figure 1: Remote Interface Interconnect

1.4 Serial ID EPROM

DS2502 is for storing board ID, connected to PIC16C65 Port C bit 6. The programming is handled through a jumper applied through connector J1. DS2502 is a signal powered, retaining the charge into a capacitor, sourced through the data line.

2.1 Alternative Power Source

The board is powered through 7.5V and 700mA (or 800mA which ever available) supply unit. After regulating the supply, it is an alternative source for the bias powered partition of the Raptor Wire Services. The bias powered block includes an NV-RAM and a PIC16C65 which are resident on the Raptor back plane.

The power source is regulated through a high frequency switching regulator based on Linear Technology LT1376. The input to the regulator circuitry is off a wall mounted adapter. The regulated output is consumed locally and 300mA are sourced to the Raptor Wire Services through a fuse and an RJ45 P1.

2.2 Power Consumption

The following is an average estimated power consumption with the board running at a base frequency of 12MHz.

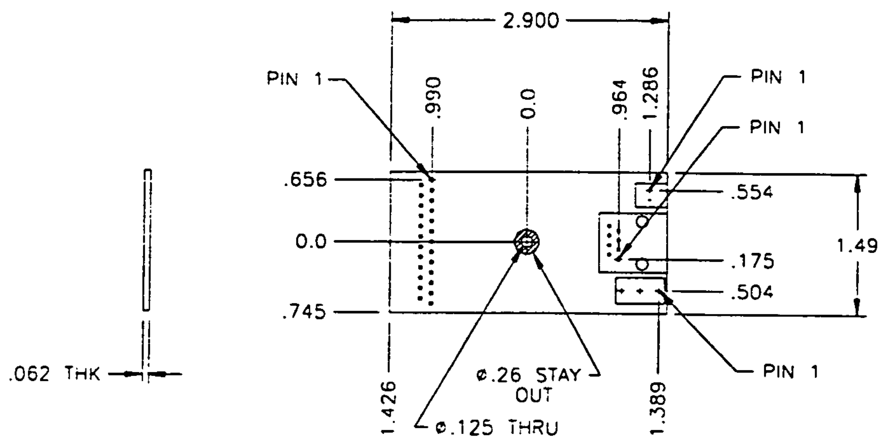
PIC16C65	Microcontroller	30mA
82B715	I2C Extender	10mA
32Kx8	SRAM	80mA
LT1133A	Transceiver	70mA
374	Latch	30mA
	Misc.	60mA

		280mA
	Alt. Source	300mA

580mA

LT1376, L1, D2, D3, etc. 1.45 Watts max.

================================



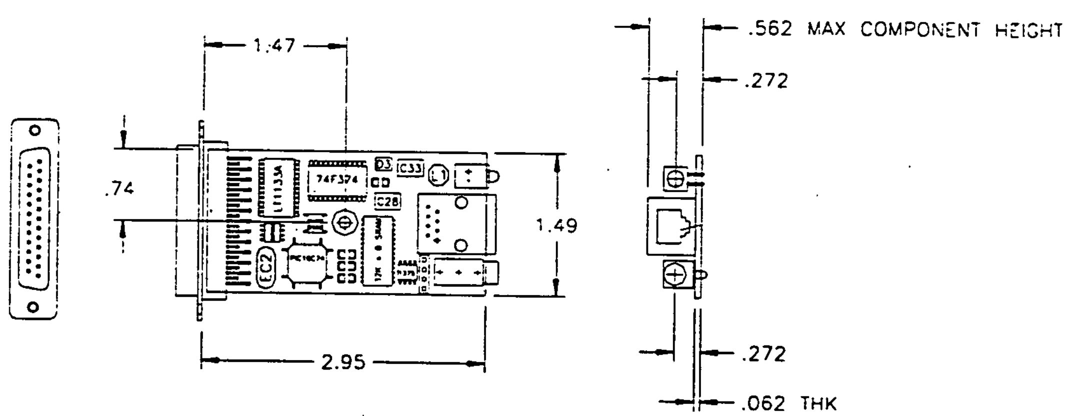


Figure 2: Mechanical Orientation

2.3 Board Layout

The board is based on controlled impedance of 60 Ohms +/- 10%, with 6 layers and test points for all signals. The width is restricted by the dimension of the RS232 due to the mounting constrains. The board is dual sided with active components kept on the top side only.

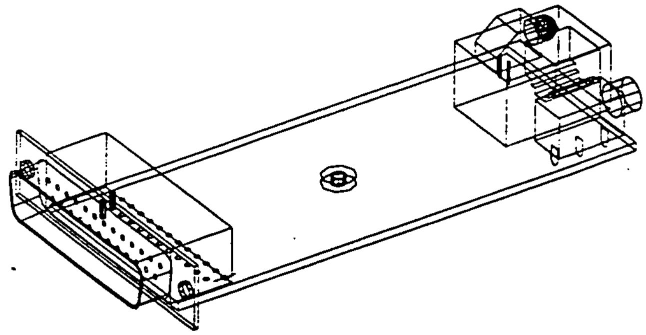
The high frequency bypass is kept with .1uf and .001uf, where the charge storage is kept by two 33uf and two 1uf capacitors.

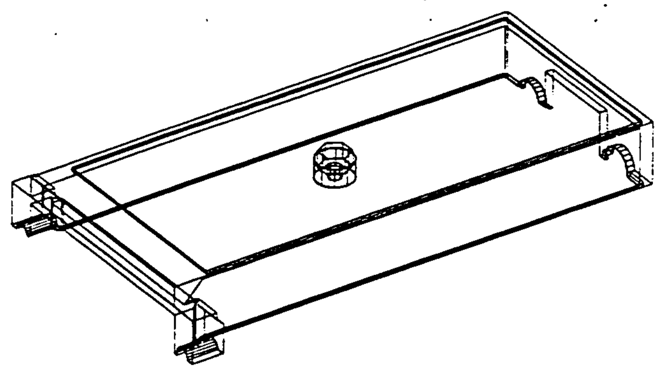
The location and mounting of the power connector and the LED are kept such that the both sides of the cabinet are identical, therefore interchangeable.

3: Enclosure

The enclosure is planned to be Injection Molded Aluminum, a side view is in figure 3. Aluminum instead of plastic is selected due to the regulator heat, and EMI shielding.

The board connects at three locations between the top and the bottom enclosures. Two locations are based on the clamp shell design at the D-Sub and the RJ45, two opposite ends of the enclosure. The third location is a mounting hole in the center of the enclosure.





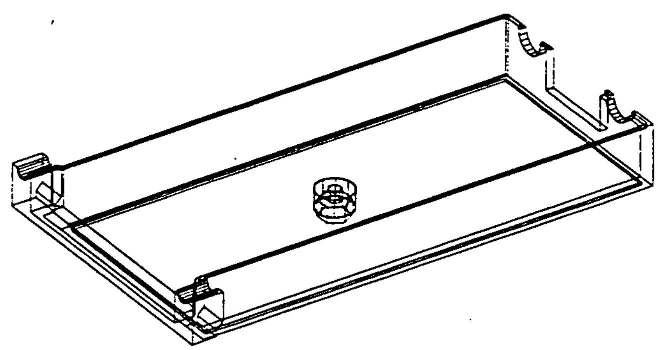


Figure 3: Board and Enclosure Isometric View

4.0 Environment

The environmental specification is based on assumptions:

The environment is Ground Fixed.
The "Quality Level of II" is used.
Bellcore Method I, Parts Count Method, Case 2 for prediction.
Burn-in time 120 hours.
Operated at 40C and 50% rated electrical stress.

4.1 Environmental Specification:

MEAN TIME BETWEEN FAILURE

2.445250e+06 Hours

This number is calculated based on the Bellcore Technical Reference TR-NWT-000332, Reliability Prediction Procedure for Electronic Equipment, Issue 4, September 1992.

ALTITUDE

Operating	-100 to 10,000 feet
Non-Operating	-100 to 40,000 feet

HUMIDITY

Operating	10% to 80% R.H., Maximum Gradient 10% per hour
Non-Operating	5% to 90% R.H., Maximum Gradient 10% per hour

TEMPERATURE (ambient)

Operating 10 to 40 degrees C Maximum Gradient 10 degrees C per hour

Non-Operating -40 to 70 degrees C Maximum Gradient 10 degrees C per hour

SHOCK

Operating

Magnitude 2 G's (peak)
Duration 11 ms
Waveform Half Sine

Non-Operating

Magnitude 10 G's (peak)
Duration 11 ms
Waveform Half Sine

VIBRATION

Operating

Frequency Range	5 to 500 Hz
Magnitude	0.010 inch peak to peak displacement
Acceleration	0.20 GÕs peak

Non-Operating

Frequency Range	5 to 500 Hz
Magnitude	0.010 inch peak to peak displacement
Acceleration	0.50 GÕs peak

DROP (PACKAGED)
ASTM D4169

ELECTRICAL

Nominal Line	115 VAC or 230 VAC @ 50/60 Hz autoranging
Line Deviation	90-130 VAC & 180-256 VAC @ 47-63 Hz
Line Transient/Surge Susceptibility	1.25 x highest rated nominal voltage or 300 Vrms, whichever is less, for 1 second.

ELECTRO-MAGNETIC COMPATIBILITY

FCC, Class A under FCC Rule 15, Subpart B, conducted and radiated.

Canadian Radio Interference Regulations, C.R.C., c.1374, Sec. 2, as amended in The Canadian Gazette, Part II, Vol. 122, No. 20, dated Sept. 28, 1988.

European EMC Directive (89/336/EEC) CISPR 22 (Class B).

IEC 801-2:1984 8 kV air discharge

IEC 801-3:1984 3 V/m, 27-500 MHz

IEC 801-4:1988 1 kV mains, 500 V other.

SAFETY AGENCIES

UL, CSA, VDE, JIS

Electrostatic Discharge

Air Discharge	2.5 to 5.0KV	no errors allowed
	5.1 to 10.0 KV	recoverable errors through system allowed
	10.0 to 20.0KV	recoverable errors through power cycling allowed
Contact Discharge	0 to 8.0KV	recoverable errors through power cycling allowed

Author: John Hammond at NetFRAMEHQ
Date: 8/29/96 1:25 PM
Priority: Normal
TO: Ahmad Nouri at NetFRAME_ENG
CC: David Werblo
CC: Michael OConnor
Subject: DSM Commands
---------------------------------- Message Contents ----------------------------------

Ahmad,

I had a look at the copy of Raptor Wire Services Architecture document and found that it was not new information for me. I have a copy of the v1.2 document. Furthermore, these four pages don't appear as "commands" to me. They seem to be more procedures.

In any event, I still like our list from the white board for FCS:

1. Power up
2. Power down
3. System Status to include system configuration (memory, processors, cannisters present, PCI cards, SCSI devices, etc.) and serial numbers where the data is available through wire services.
4. Read the Flight Recorder
5. Reset System. It would be nice to have a cold and warm reboot but if not, lets emulate the reset switch on the front panel for FCS.

Thanks,

John

EXHIBIT E

Raptor Wire Service Architecture

Version 1.3

October 3, 1996

Prepared for
NetFrame Raptor Implementation Group

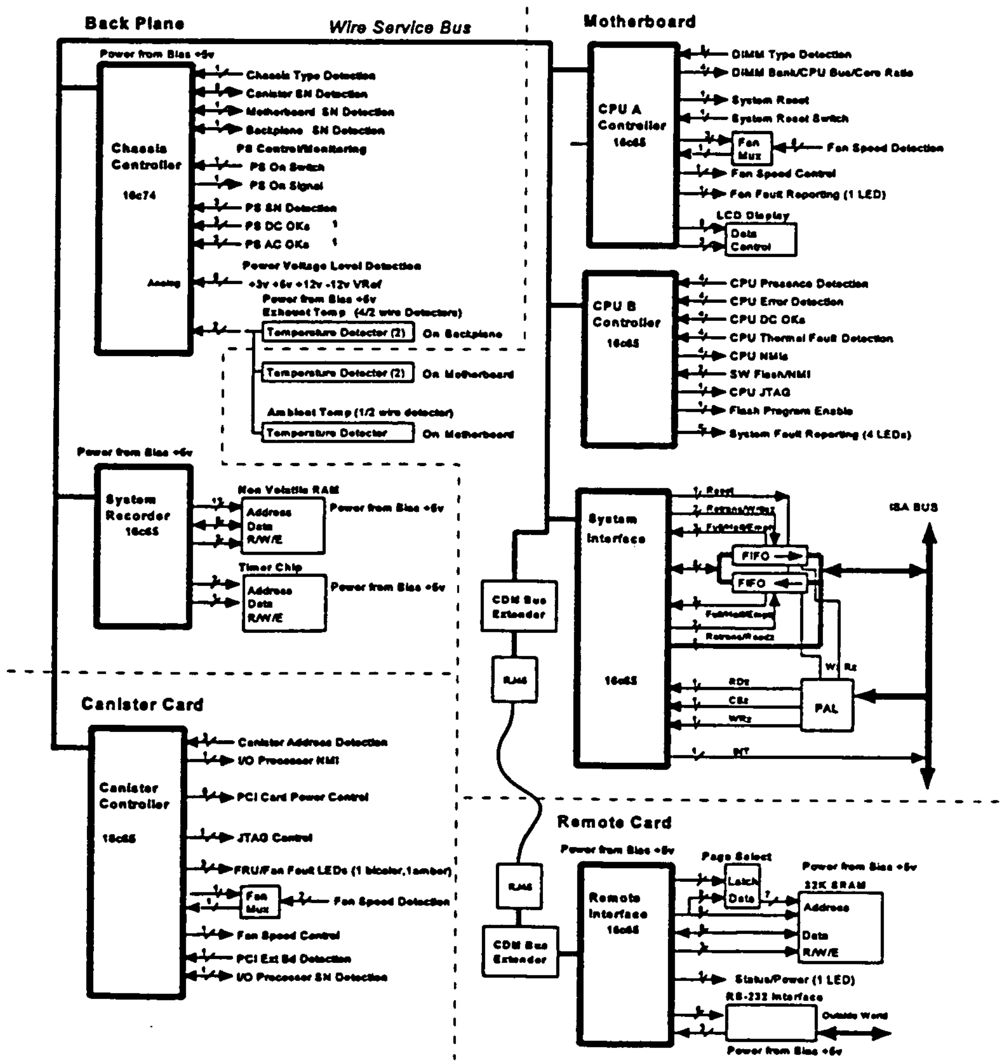
by
Karl Johnson (KJ)

EXHIBIT F

Introduction

"Wire Service" is the code name for the Raptor project system control, diagnostic and maintenance bus (formerly known as the CDM bus). Raptor is a completely "fly by wire" system - no switch, indicator or other control is directly connected to the function it monitors or controls. Instead, all the control and monitoring connections are made by the network of processors that comprise the "Wire Service" for the system. The processors are Microchip PIC processors and the network is a 400 kbps I^2C serial bus. A limited understanding of I^2C protocol is a prerequisite for understanding Wire Service protocols (See "The I^2C-bus and how to use it" - Philips Semiconductor, Jan 1992). Control on this bus is distributed, each processor can be either a master or a slave and can control resources on itself or any other processor on the bus.

Wire Service Hardware Block Diagram



Write Byte Array Message
Request

Slave Address	Type/RW	Start Addr LSB	Start Addr MSB	Length N	Array Data 1	. . .	Array Data N

Response

Slave Address	Length 0	Status 0/Success

Log Type

The Log data type is to be used for logging byte strings in circular log buffer. It is used to record system events in the NVRAM system log.

Read Log Message
Request

Slave Address	Type/RW	Log Addr LSB	Log Addr MSB	Request 1-255

Response

Slave Address	Length N+6	Log Time LSB	Log Time	Log Time	Log Time MSB	Log Addr LSB	Log Addr MSB
Log Data Byte 1	. . .	Log Data Byte N	Status 0/Success				

Write Log Message
Request

Slave Address	Type/RW	N/A	N/A	Length N(1-249)	Log Data Byte 1	. . .	Log Data Byte N

Response

Slave Address	Length 0	Status 0/Success

Note: The maximum number of bytes that can be written in a log entry is 249, because the log processor adds 6 bytes of ID and timestamp to the beginning of the message.

The addressing of log entries has some special characteristics:

- Reading address 65534 (0xfffe) is also special - It represents the address of the earliest available entry.

- Reading address 65533 (0xfffd) is special - It represents the address of next message in sequence from the last message read from the log.
- The address of real log entries wraps at 65279 (0xfeff). The next sequential entry after 65279 is 0.
- Onlyt the two special addresses above are recognized in reading the log. The address of is ignored on write and the next available entry is written.
- To read the entire log in forward time order, read entry at address 65534. This returns the first log entry. Next read from address 65533 to get the next sequential log entry. Repeat the last step until status indicates failure.
- To keep a complete external copy of the log, first read the entire log in forward time order. Then periodically read from the next valid entry (65533) to the end and add that to the external copy.

Note: Multiple simultaneous readers of the log will interfere with each other. This may be detected by the fact that each reader will see a stream of entries that are missing some log entry sequence numbers. To recover, the log must be read from the oldest entry again. If multiple readers are anticipated, a byte in the byte array area should be selected as a semaphore and the lock byte data type can be used to synchronize access to the log.

Note:Log messages can also appear out of sequence if the log is being filled rapidly enough that old messages must be discarded before they are read. In this case the log processor returns the oldest currently available entry for the next entry read.

There are also conventions for the Log Data portion of the message or response. These are conventions and are not enforced by the Wire Service logging function. The conventions are as follows:

Log Data Byte 1: Severity Level Byte
0x00 - Unknown
0x10 - Informational
0x20 - Warning
0x30 - Error
0x40 - Severe/Fatal Error

Log Data Byte 2: Source/Encoding Byte - which entity logged the entry in the 4 high bits
0x00 - Wire Service Internal
0x10 - Onboard Diagnostics
0x20 - External Diagnostics
0x30 - BIOS
0x40 - Time Synchronizer
0x50 - Windows
0x60 - Windows/NT
0x70 - NetWare
0x80 - OS/2
0x90 - UNIX
0xA0 - VAX/VMS
- which type of encoding of the message is used in the 4 low bits of the byte.

0x00 - Binary
0x01 - ASCII
0x02 - Unicode

Thus an external diagnostics ASCII error message would have a severity and source/encoding byte values of 0x30 and 0x21.

For binary encoded messages, additional conventions apply:

Log Data Byte 3 and 4 are used as the LSB and MSB of a 16 bit Message Identifier. Each source group (OS's, BIOS, etc..) that wishes to log binary messages must maintain a file in a common format containing the definition of ALL possible binary messages from their source. This file contains the Identifier value and formatting string and any descriptive comments for each message. Identmust be maintained by each possible message source group and supplied as a file to any requestor. (Obsolete Message Identifiers MAY NOT be reused for old version compatability reasons) Log Data Bytes 5 and beyond are message arguments in format list order.

An example binary message definition file entry for the BIOS might be:
00045 "DIMM configuration bad - row %db - Types %3.3db %3.3db %3.3db %3.3db"

> Note: Most "C" printf formatting items will work except that the final letter of a numeric formatting sequence must be b (byte) s (short) or l (long) to how many bytes of the log message data it takes. Strings take up to the first null (zero terminated).

Event Type

The event data type is to be used for alerting external interfaces of events in the Wire Service network. Event memory is organized as a bit vector. Each bit in the vector represents a particular type of event and there are a minimum of 16 bits in the vector.. Writing an event sets the bit representing the event in the bit vector. Whenever any bit in the event bit vector is non-zero, the interface indicates that events are present..

Reading the event type returns one or more event ID's, depending on request length and events actually pending in the interface. Once an event ID is read, the corresponding event bit is automatically cleared internally.

Request

Slave Address	Type/RW	N/A	N/A	Request 1-255

Response

Slave Address	Length N(1-32)	Event ID	. . .	Event ID	Status 0/Success

Write Event Message

Request

Slave Address	Type/RW	N/A	N/A	Length 1	Event ID (1-15)

Response

Slave Address	Length 0	Status 0/Success

Possible Event Types:
CPU Status Change
Power Status Change
Canister Status Change
Fan Status Change
Communications Queue Event

Wire Service Remote Interface Serial Protocol

The Wire Service Remote Serial protocol is used to communicate Wire Service messages across a serial link from a Wire Service Remote Interface processor attached to a Raptor to a Wire Service Remote management processor. It encapsulates Wire Services messages in a transmission envelope to provide error free communications and link security.

In order to be easily processed by the remote interface processor, the remote interface protocol is based on the concept of byte stuffing, That is certain byte values in the data stream always have a particular meaning and if that value must be transmitted as by the underlying application as data, it must be transmitted as a two byte sequence.

The bytes that have special meaning in the protocol are:
SOM - Start of a message
EOM - End of a message
SUB - The following byte in the data stream must be substituted before processing
INT - Event Interrupt

If any of these byte values occur as data in a message, a two byte sequence must be substituted for the byte. That sequence is a byte with the value of SUB followed by a byte with the value of the original byte incremented by one. For example if a SUB byte were to occcur in a message it would appear on the serial data stream as a SUB followed by a byte whose value is SUB+1.

Just as with Wire Service messages there are two classes of messages: 1) Requests - sent by remote managment systems (PC's) to the Wire Service remote interface and 2) Responses - returned to the requester by the Wire Service interface. The formats for the messages are as follows:

Request

SOM	Seq. #	TYPE	Data	...	Check	EOM

Response

SOM	Seq. #	STATUS	Data	...	Check	EOM

Event Interrupt

INT

Where:
SOM/EOM - Special data byte values marking the start and end of messages.
Seq # - A one byte sequence number that increments on each request and is copied in the response.

TYPE - One of the following types of requests:

- IDENTIFY- Request the remote interface to send back identification information about the system to which it is connected. This also resets the next expected sequence number. This message type does not require security authorization to be established first.
- SECURE - Establish secure authorization on the serial link by checking password security information provided in the message with the Wire Service system password.
- UNSECURE - Clear security authorization on the link and attempt to disconnect it. (Requires security authorization to have been established)
- MESSAGE - Take the data portion of the message and pass it to Wire Service for execution. The response from Wire Service is sent back in the data portion of the response. (Requires security authorization to have been established)
- POLL - Query the status of the remote interface. This is generally used to determine if an event is pending in the remote interface.

STATUS - One of the following response status values:

- OK - Everything relating to communication with the remote interface was OK
- OK_EVENT - Everything relating to communication with the remote interface was OK and there is one or more events pending in the remote interface.
- SEQUENCE - The sequence number of the request was not the current sequence number (retransmission request) or the next expected sequence number (new request). Sequence numbers may be reset by a IDENTIFY message.
- CHECK - The check byte in the request message was received incorrectly.
- FORMAT - Something about the format of the message was incorect. Most likley, the type field has an invalid value.
- SECURE - The message requires that security authorization be in effect, or if the message was of type SECURE, the security check failed.

Check - A message integrety check byte. Currently the value is 256 - the sum modulo 256 of all previous bytes in the message. (i.e. Adding all bytes in the message up to and including the check byte should produce a result of zero (0))

INT - A special one byte message sent by the remote interface when it detects the transistion from no events pending to one or more events pending. This message can be used to trigger reading events from the remote interface. Events should be read until the return status changes from OK_EVENT to just OK.

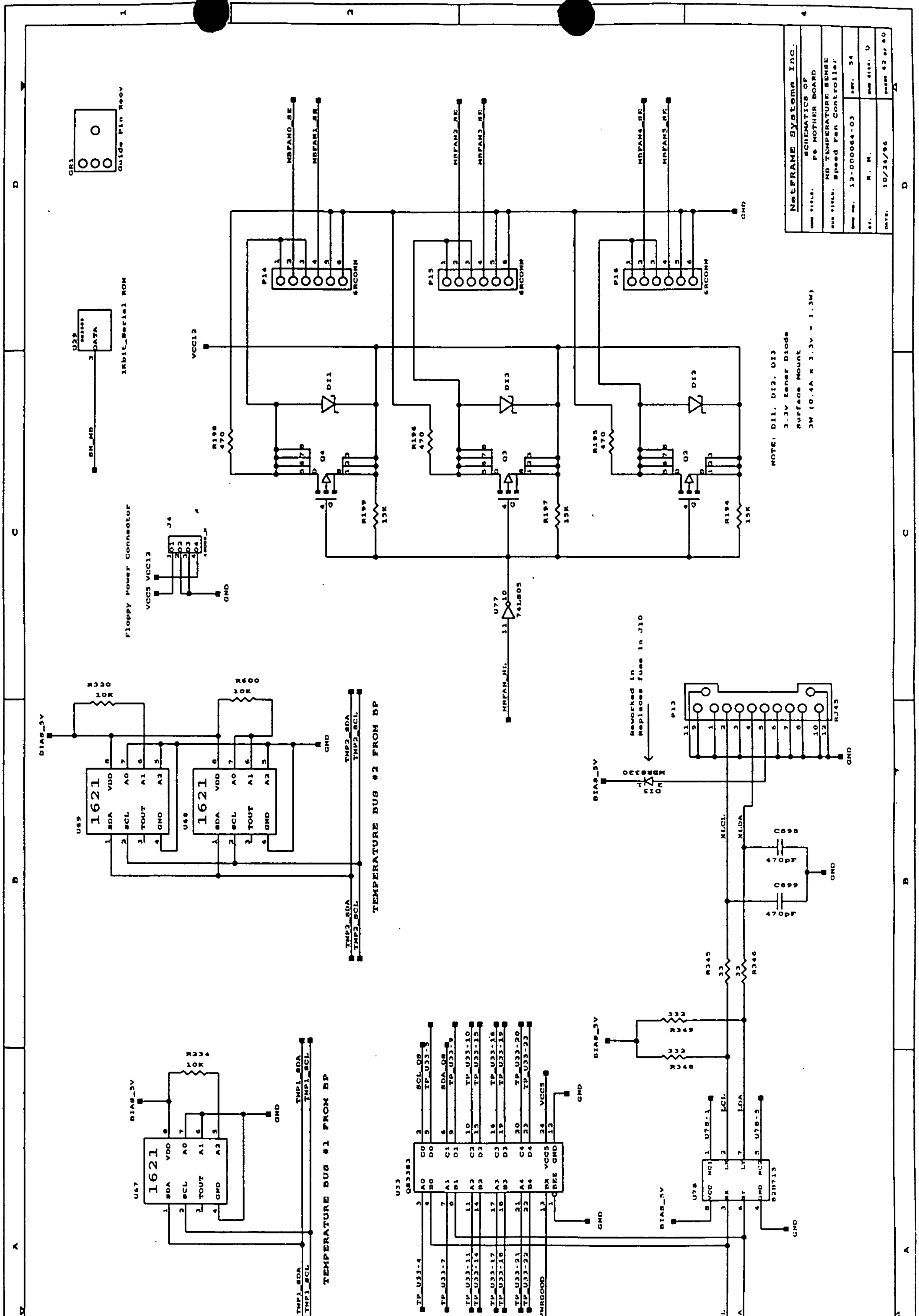

NetFRAME Systems Inc.
TEMPERATURE BUS #2 FROM BP
TEMPERATURE BUS #1 FROM BP
Floppy Power Connector
1Kbit_Serial ROM
Guide Pin Recv

EXHIBIT G

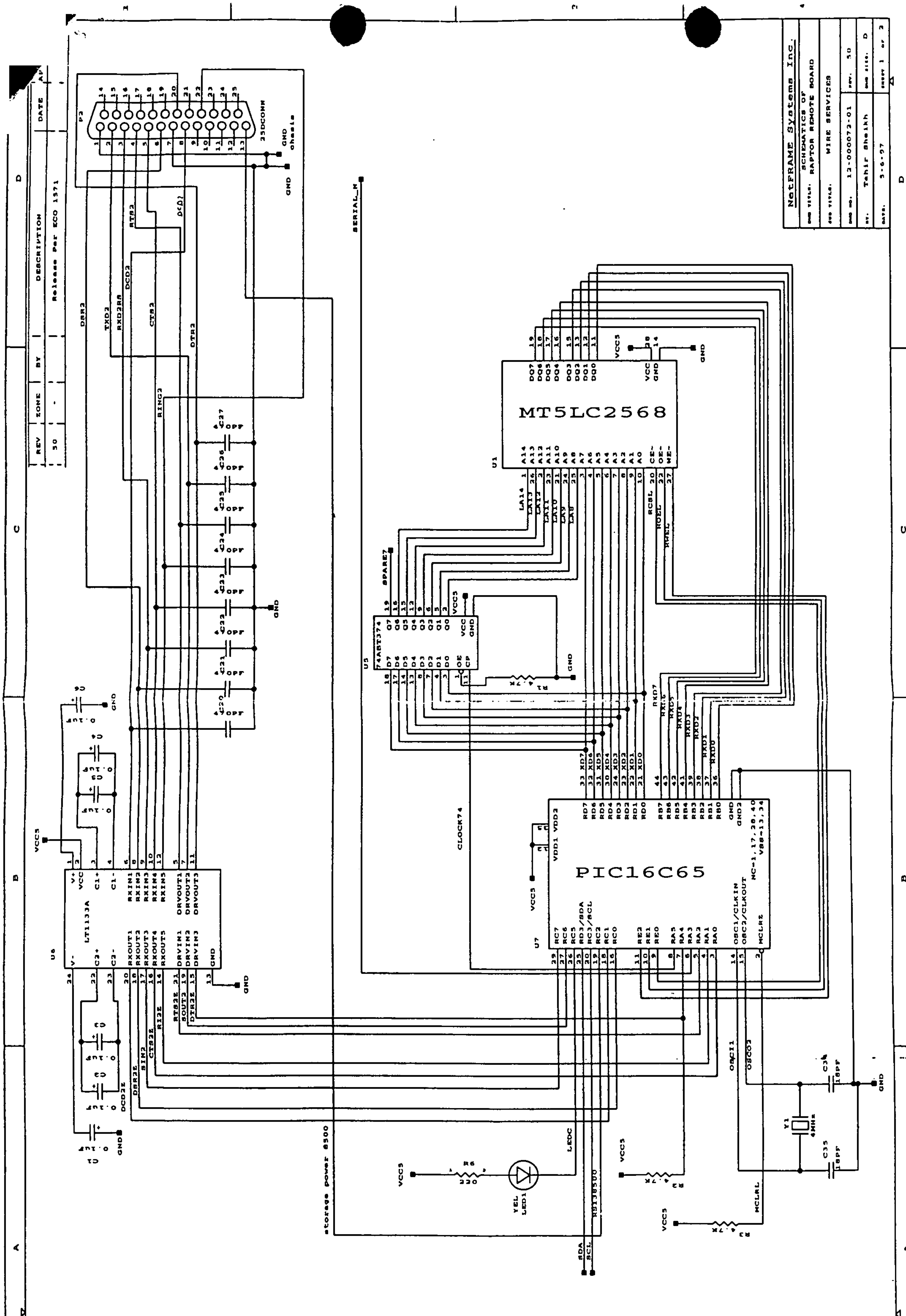
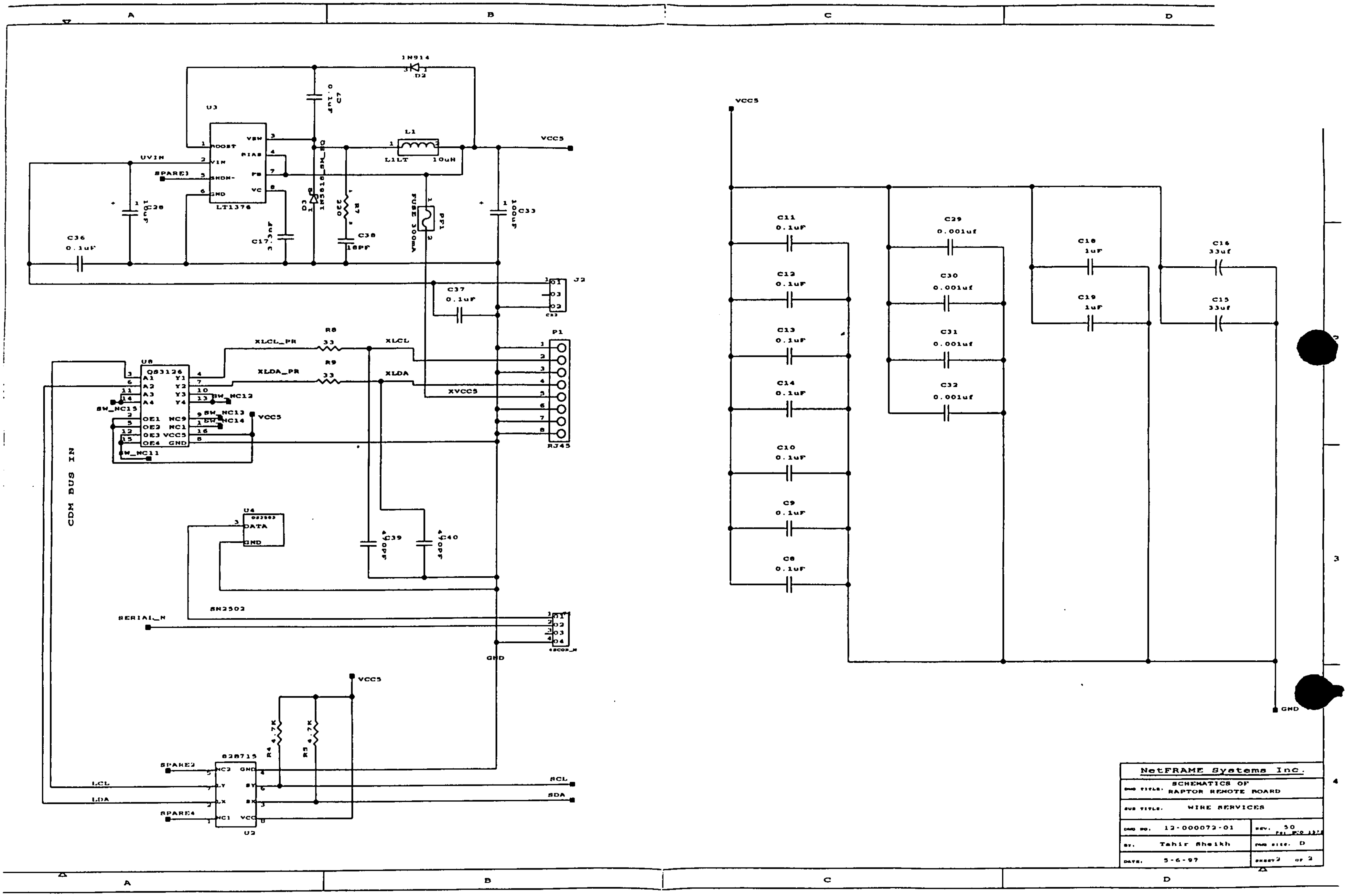

NetFRAME Systems Inc.
SCHEMATICS OF RAPTOR REMOTE BOARD
WIRE SERVICES
12-000072-01
Tahir Shaikh
5-6-97
Release Per ECO 1571
MT5LC2568
PIC16C65
LT1133A
74ABT374
25DCONN
storage power 8500
SERIAL_M
CLOCK74
SPARE7

EXHIBIT H


NetFRAME Systems Inc.
DWG TITLE: SCHEMATICS OF RAPTOR REMOTE BOARD
SUB TITLE: WIRE SERVICES
DWG NO. 12-000072-01
REV. 50
BY: Tahir Sheikh
DWG SIZE: D
DATE: 5-6-97
SHEET 2 OF 2
CDM BUS IN
U3
LT1376
U6
QS3126
U4
U2
SN2502
VCC5
GND
UVIN
SPARE1
SPARE2
SPARE4
SERIAL_N
SCL
SDA
LCL
LDA
XLCL_PR
XLDA_PR
XLCL
XLDA
XVCC5
RJ45
P1
J2
L1
L1LT
10uH
1N914
D2
D3
PF1
FUSE 300mA
R7
220
R8
R9
33
R4
R5
4.7K
C36
0.1uF
C28
10uF
C7
0.1uF
C17
3.3nF
C38
18PF
C33
100uF
C37
0.1uF
C39
470PF
C40
470PF
C11
C12
C13
C14
C10
C9
C8
0.1uF
C29
C30
C31
C32
0.001uf
C18
C19
1uF
C16
C15
33uf